

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 06-282417

(43)Date of publication of application : 07.10.1994

(51)Int.Cl.

G06F 7/50

(21)Application number : 05-071406

(71)Applicant : NEC CORP

(22)Date of filing : 30.03.1993

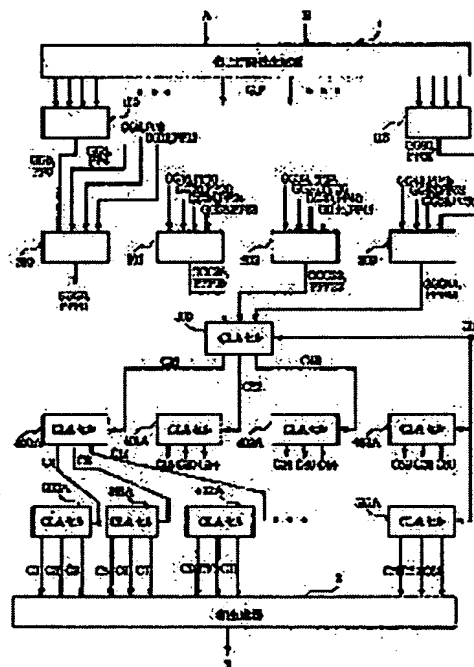
(72)Inventor : OKAMOTO FUYUKI

(54) ADDING CIRCUIT

(57)Abstract:

PURPOSE: To shorten the delay time of a critical path where a carry is propagated and speed up arithmetic by providing a logic circuit which makes the delay time of the least significant digit carry propagation path of CLA cells in a 1st and a 2nd stage layer shorter than the delay time of CLA cells of a 3rd stage.

CONSTITUTION: In addition to a carry function generator 1, block carry function generators 100-115, and 200-203, and the CLA cell 300 in the 3rd stage layer, and a sum generator 2, this circuit is equipped with the CLA cells 400A-403A of the 2nd stage as 2nd type CLA cells and CLA cells 500A-515A of the 1st stage layer. The 2nd type using a two-input NAND gate is shorter in the delay time from the input of a carry input C_{in} from a low order layer to the acquisition of a carry output to a high-order layer and faster than a 1st type CLA cell which uses a 4-input NAND gate. Therefore, the 2nd type CLA cells are used for the 1st and 2nd stage layers to reduce the delay of the carry input to the lowest order block and the operation is speeded up.



LEGAL STATUS

[Date of request for examination] 30.03.1993

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number] 2067568

[Date of registration] 10.07.1996

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

(19)日本国特許庁 (J P)

(12) 公 開 特 許 公 報 (A)

(11)特許出願公開番号

特開平6-282417

(43)公開日 平成 6 年(1994)10月 7 日

(51)Int.Cl.⁵

G 0 6 F 7/50

識別記号

庁内整理番号

F I

技術表示箇所

G 9291-5B

審査請求 有 請求項の数 3 O L (全 10 頁)

(21)出願番号 特願平5-71406

(22)出願日 平成 5 年(1993) 3 月30日

(71)出願人 000004237

日本電気株式会社

東京都港区芝五丁目 7 番 1 号

(72)発明者 岡本 冬樹

東京都港区芝五丁目 7 番 1 号日本電気株式
会社内

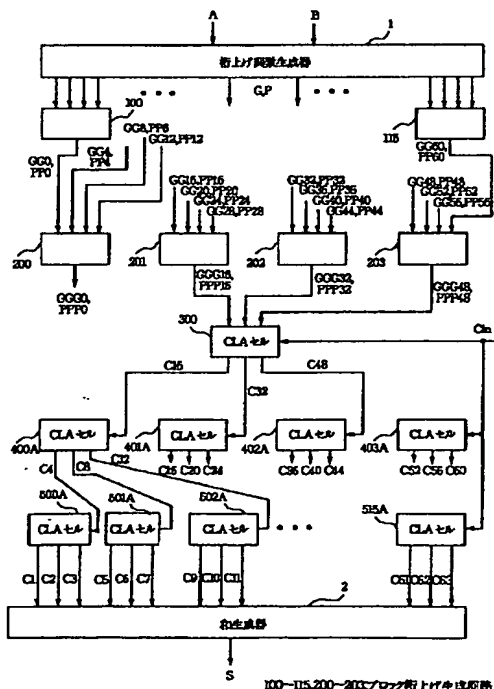
(74)代理人 弁理士 京本 直樹 (外 2 名)

(54)【発明の名称】 加算回路

(57)【要約】

【目的】従来よりも高速な加算回路を提供する。

【構成】第1, 第2 階層のCLAセル400A~402
Aおよび500A~515Aは、桁上げ入力C_{in}の伝
播経路の遅延時間が第3 階層のCLAセル300の上記
遅延時間よりも小さい論理素子E25, E27, E2
6, E28, E8, E12から成る2入力NANDゲー
トの2 段構成の論理回路を備える。



1

【 特許請求の範囲】

【 請求項1 】 それぞれ $m \times n$ (m, n は整数) ビットの加数と被加数とから第1 階層の桁上げ発生関数および桁上げ伝播関数とを生成する第1 階層の桁上げ関数生成手段と、

前記第1 階層の桁上げ発生関数と桁上げ伝播関数との供給を受けこれら第1 階層の桁上げ発生関数と桁上げ伝播関数とを m ビット毎にブロック化した第2 階層のブロック桁上げ発生関数とブロック桁上げ伝播関数とを生成する n 個の第2 階層のブロック桁上げ関数生成手段と、

前記第2 階層の桁上げ発生関数と桁上げ伝播関数との供給を受けこれら第2 階層の桁上げ発生関数と桁上げ伝播関数とを m ビット毎にブロック化した第3 階層のブロック桁上げ発生関数とブロック桁上げ伝播関数とを生成する n/m 個の第3 階層のブロック桁上げ関数生成手段と、

前記第3 階層のブロック桁上げ発生関数とブロック桁上げ伝播関数と最下位ビットに対する桁上げ入力である最下位桁上げ入力との供給を受け $m-1$ 個の予め定めた第1 のビット番号の桁の桁上げである第1 の桁上げ出力を算出する第3 階層の第1 の桁上げ先見(C L A) セルと、

前記第1 の桁上げと前記第2 階層の桁上げ発生関数と桁上げ伝播関数と前記最下位桁上げとの供給を受けそれぞれ $m-1$ 個の予め定めたビット番号の桁の桁上げである第2 の桁上げ出力を算出する n/m 個の第2 階層の第2 のC L A セルと、

前記第2 の桁上げ出力と前記第1 階層の桁上げ発生関数と桁上げ伝播関数と前記最下位桁上げ入力との供給を受け全ての桁の桁上げである第3 の桁上げ出力を算出する n 個の第1 階層の第3 のC L A セルと、

前記第3 の桁上げ出力の供給を受け和を生成する和生成器とを備える加算回路において、

前記第2 , 第3 のC L A セルが前記最下位桁上げ入力を伝播する最下位桁上げ伝播経路の遅延時間を前記第1 のC L A セルの前記遅延時間よりも小さくする構成の論理回路を備えることを特徴とする加算回路。

【 請求項2 】 前記論理回路の前記最下位桁上げ伝播経路が縦続接続された2 段の2 入力NAND回路から成ることを特徴とする請求項1 記載の加算回路。

【 請求項3 】 前記第2 および第3 のC L A セルが、第1 , 第2 および第3 の桁上げ伝播関数の否定論理積をとり第1 の論理値を出力する3 入力NANDゲートである第1 の論理素子と、

前記第1 および第2 の桁上げ伝播関数と第3 の桁上げ発生関数との否定論理積をとり第2 の論理値を出力する3 入力NANDゲートである第2 の論理素子と、

前記第1 の桁上げ伝播関数と第2 の桁上げ発生関数との否定論理積をとり第3 の論理値を出力する2 入力NANDゲートである第3 の論理素子と、

2

第1 の桁上げ発生関数を反転し第4 の論理値を出力するインバータである第4 の論理素子と、

前記第1 および第2 の桁上げ伝播関数の否定論理積をとり第5 の論理値を出力する2 入力NANDゲートである第5 の論理素子と、

前記第2 の桁上げ伝播関数と前記第3 の桁上げ発生関数との否定論理積をとり第6 の論理値 F_6 を出力する2 入力NANDゲートである第6 の論理素子と、

前記第2 の桁上げ発生関数を反転し第7 の論理値を出力するインバータである第7 の論理素子と、

前記第3 の桁上げ伝播関数 P_2 と最下位桁上げ入力との否定論理積をとり第8 の論理値を出力する2 入力NANDゲートである第8 の論理素子と、

前記第3 の桁上げ発生関数を反転し第9 の論理値 F_9 を出力するインバータである第9 の論理素子と、

前記第1 , 第2 および第3 の論理値の否定論理積をとり第10 の論理値を出力する3 入力NANDゲートである第10 の論理素子と、

前記第10 および第6 の論理値の否定論理積をとり第11 の論理値を出力する2 入力NANDゲートである第11 の論理素子と、

前記第10 の論理値と前記最下位桁上げ入力との否定論理積をとり第12 の論理値を出力する2 入力NANDゲートである第12 の論理素子と、

前記第11 の論理値と前記最下位桁上げ入力との否定論理積をとり第13 の論理値を出力する2 入力NANDゲートである第13 の論理素子と、

前記第4 および第11 の論理値の否定論理積をとり第14 の桁上げ出力を出力する2 入力NANDゲートである第14 の論理素子と、

前記第7 および第13 の論理値の否定論理積をとり第2 の桁上げ出力を出力する2 入力NANDゲートである第15 の論理素子と、

前記第8 および第9 の論理値の否定論理積をとり第3 の桁上げ出力を出力する2 入力NANDゲートである第16 の論理素子とを備えることを特徴とする請求項1 記載の加算回路。

【 発明の詳細な説明】

【 0 0 0 1 】

【 産業上の利用分野】 本発明は加算回路に関し、特に多ビットの演算を4 または8 ビットのブロックに分割しこの4 または8 ビット単位のC L A (桁上げ先見) セルを複数個縦続接続して加算の高速化を図った加算回路に関する。

【 0 0 0 2 】

【 従来の技術】 C L A 加算器は、演算すべき n 桁のそれぞれの桁毎に、その桁より下位の桁から別に求めた桁上げを用いて演算する方式であり、加算全体の演算時間を決定する桁上げ伝播時間が全加算器を用いる場合に比較して短縮されるので、高速化の手法として広く使われて

いる。多ビットの演算の場合、4 または8 ビットのブロックに分割しこの4 または8 ビット単位のCLA加算器 (CLAセル) を複数個縦続接続して所要ビットの加算回路を構成することが多い。

【0003】4 ビットのCLAセルを用いたブロック構成の従来のこの種の6 4 ビットの加算回路の一例を示す図3を参照すると、この従来の加算回路は、それぞれ6 4 ビットの入力AとBとから桁上げ発生関数Gと桁上げ伝播関数Pとを生成する第1 階層の桁上げ関数生成器1と、第2 階層、第3 階層のそれぞれのブロック桁上げ関数生成器100 ~ 115, 200 ~ 203と、第3 階層の4 ビットのCLAセル300と、各々が4 ビットの第2 階層、第1 階層のCLAセル400 ~ 403, 500 ~ 515と、CLAセル500 ~ 515の出力から和Sを生成する和生成器2とを備える。

【0004】桁上げ関数生成器1が inputs A, B から生成する桁上げ発生関数Gの各々のビットGiは、Aの各々のビットAiとBの各々のビットBiとから次式で求め*

$$GG_i = G_i + P_i G_{i+1} + P_i P_{i+1} G_{i+2} + P_i P_{i+1} P_{i+2} G_{i+3} \quad \dots (1)$$

(i=0,4,8,12,16,20,24,28,32,36,40,44,48,52,56,60)

【0011】GGiは、第i ~ 第i + 3 ビットを一まとめた1つのブロックとみなし、このブロック内から上位(第i - 1 ビット)ブロックへの桁上げが発生する※

$$PP_i = P_i P_{i+1} P_{i+2} P_{i+3} \quad \dots (2)$$

(i=0,4,8,12,16,20,24,28,32,36,40,44,48,52,56,60)

【0013】PPiは、GGiと同様、第i ~ 第i + 3 ビットから成るブロックの最下位(第i + 3 ビット)へ入力された桁上げがそのままブロック内を伝播して上位ブロックへ伝播するとき真となる。なお、今後便宜上これら関数GGとPPをそれぞれ第2 階層のブロック桁上げ発生関数、第2 階層のブロック桁上げ伝播関数と呼ぶ。

【0014】さらに、桁上げ発生関数GGと桁上げ伝播★

$$GGG_i = GG_i + PP_i GG_{i+4} + PP_i PP_{i+4} GG_{i+8} + PP_i PP_{i+4} PP_{i+8} GG_{i+12}$$

$$PPP_i = PP_i PP_{i+4} PP_{i+8} PP_{i+12} \quad \dots (3)$$

(i=0,16,32,48)

【0016】GGGiは、第i ~ 第i + 15 ビットを1つのブロックとみなし、このブロック内から上位ブロックへの桁上げが発生するとき真となる。PPPiは、同様に、第i ~ 第i + 15 ビットから成るブロックの最下位(第i + 15 ビット)へ入力された桁上げがそのままブロック内を伝播して上位ブロックへ伝播するとき真となる。なお、今後これら関数GGGとPPPをそれぞれ第3 階層のブロック桁上げ発生関数、第3 階層のブロッ

$$C_{15} = GGG_{15} + PPP_{15} GGG_{32} + PPP_{15} PPP_{32} GGG_{48} + PPP_{15} PPP_{32} PPP_{48} C_{in}$$

$$C_{32} = GGG_{32} + PPP_{32} GGG_{48} + PPP_{32} PPP_{48} C_{in}$$

$$C_{48} = GGG_{48} + PPP_{48} C_{in} \quad \dots (4)$$

*られる。

【0005】

$$G_i = A_i \cdot B_i$$

【0006】Giは、AとBとの加算において第i 番目のビットで桁上げが発生するとき真である。また、桁上げ伝播関数Pの各々のビットPiは、Aの各々のビットAiとBの各々のビットBiとから次式で求められる。

【0007】

$$P_i = A_i \oplus B_i$$

【0008】Giは、AとBとの加算において第i 番目のビットで桁上げが伝播するとき真である。

【0009】桁上げ発生関数Gと桁上げ伝播関数Pとの供給を受け、第2 階層のブロック桁上げ関数生成器100 ~ 115が、これら関数G, Pを4 ビット単位のブロックにまとめてブロック桁上げ発生関数GGとブロック桁上げ伝播関数PPとを出力する。ここでブロック桁上げ発生関数GGは次式で定義される。

【0010】

※とき真となる。また、ブロック桁上げ伝播関数PPは次式で定義される。

【0012】

★関数PPとの供給を受け、第3 階層のブロック桁上げ関数生成器200 ~ 203が、これら関数GGとPPとを4 ビットずつブロック化してブロック桁上げ発生関数GGとブロック桁上げ伝播関数PPPとを出力する。ここでこれら関数GGGおよびPPPは次式で定義される。

【0015】

ク桁上げ伝播関数と呼ぶ。

【0017】上述のようにして求められたこれら第3 階層のブロック桁上げ発生／伝播関数GGG, PPPと、最下位ビットに対する桁上げ入力Cinとの供給を受け、第3 階層のCLAセル300が第16, 32, 48 ビットのそれぞれからの次式で示す桁上げC16, C32, C48を算出する。

【0018】

5

【 0 0 1 9 】式(4) の計算を実現する第1 型のCLAセルの回路図である図4 を参照すると、この第1 型のCLAセルは、関数P 0 ～P 2 と桁上げ入力C_{i n} との否定論理積をとり論理値F 1 を出力する4 入力NANDゲートである論理素子E 1 と、関数P 0 , P 1 , G 2 の否定論理積をとり論理値F 2 を出力する3 入力NANDゲートである論理素子E 2 と、関数P 0 , G 1 の否定論理積をとり論理値F 3 を出力する2 入力NANDゲートである論理素子E 3 と、関数G 0 を反転し論理値F 4 を出力するインバータである論理素子E 4 と、関数P 1 , P 2 と桁上げ入力C_{i n} との否定論理積をとり論理値F 5 を出力する3 入力NANDゲートである論理素子E 5 と、関数P 1 , G 2 の否定論理積をとり論理値F 6 を出力する2 入力NANDゲートである論理素子E 6 と、関数G 1 を反転し論理値F 7 を出力するインバータである論理素子E 7 と、関数P 2 と桁上げ入力C_{i n} との否定論理積をとり論理値F 8 を出力する2 入力NANDゲートである論理素子E 8 と、関数G 2 を反転し論理値F 9 を出力するインバータである論理素子E 9 と、論理値F 1 ～F 4 の否定論理積をとり論理値C 0 を出力する4 入力NANDゲートである論理素子E 1 0 と、論理値F 5 ～F 7 の否定論理積をとり論理値C 1 を出力する3 入力NANDゲートである論理素子E 1 1 と、論理値F 8 , F 9 の否定論理積をとり論理値C 2 を出力する2 入力NANDゲートである論理素子E 1 2 とを備える。

【 0 0 2 0 】図4 において、関数P 0 が第3 階層のブロック桁上げ伝播関数PPP 1 6 に、P 1 がPPP 3 2 に、P 2 がPPP 4 8 に、関数G 0 が第3 階層のブロック桁上げ発生関数GGG 1 6 に、G 1 がGGG 3 2 に、G 2 がGGG 4 8 にそれぞれ対応している。

【 0 0 2 1 】次に、CLAセル3 0 0 から供給される桁上げC 1 6 と関数GG 4 , PP 4 , GG 8 , PP 8 , G G 1 2 , およびPP 1 2 とを用いて、第2 階層のCLAセル4 0 0 が桁上げC 4 , C 8 , C 1 2 を計算する。同様に、CLAセル4 0 1 が桁上げC 3 2 と関数GG 2 0 , PP 2 0 , GG 2 4 , PP 2 4 , GG 2 8 , およびPP 2 8 とを用いて桁上げC 2 0 , C 2 4 , C 2 8 を計算する。同様に、CLAセル4 0 2 が桁上げC 4 8 と関数GG 3 6 , PP 3 6 , GG 4 0 , PP 4 0 , GG 4 4 , およびPP 4 4 とを用いて桁上げC 3 6 , C 4 0 , C 4 4 を計算する。同様に、CLAセル4 0 3 が桁上げ入力C_{i n} と関数GG 5 2 , PP 5 2 , GG 5 6 , PP 5 6 , GG 6 0 , およびPP 6 0 とを用いて桁上げC 5 2 , C 5 6 , C 6 0 を計算する。ここで、各々の桁上げC_i の計算式は式(4) と同一の形であるので、これらCLAセル4 0 0 ～4 0 1 は、CLAセル3 0 0 と同様の第1 型のCLAセルを用いる。

【 0 0 2 2 】次に、第2 階層のCLAセル4 0 0 ～4 0 3 から供給される桁上げC 4 , C 8 , C 1 2 , C 1 6 , C 2 0 , C 2 4 , C 3 6 , C 4 0 , C 4 4 , C 5 2 , C

6

5 6 , C 6 0 と関数G , P と桁上げ入力C_{i n} とを用いて、第1 階層のCLAセル5 0 0 ～5 1 5 が桁上げC 0 , C 1 , C 2 , …を計算する。これらCLAセル5 0 0 ～5 1 5 も、同様に第1 型のCLAセルを用いる。

【 0 0 2 3 】上述のように、得られた全部の桁上げC 0 ～C 6 3 から、和生成器2 は最後に和S を算出する。この和S の各々のビットS_i は次式で表される。

【 0 0 2 4 】

$$S_i = P_i \oplus C_{i+1}$$

【 0 0 2 5 】上述のように、CLAセル3 0 0 は第3 階層のブロック桁上げ発生／伝播関数GGG , PPP の供給を受け、桁上げC 1 6 , C 3 2 , C 4 8 を計算し、第2 階層のCLAセル4 0 0 ～4 0 3 に供給する。このとき、これら第1 型のCLAセル4 0 0 ～4 0 2 の内部において、それぞれ入力された桁上げC 1 6 , C 3 2 , C 4 8 が4 入力NANDゲートである論理素子E 1 およびE 1 0 を経由するパスが最も遅延時間が大きいクリティカルパスである。また、上述の計算結果である桁上げは、次の第1 階層のCLAセル5 0 0 ～5 1 5 の桁上げ入力となり、同様に、CLAセル5 0 0 ～5 1 5 の各々の内部の4 入力NANDゲート論理素子E 1 , E 1 0 を経由して桁上げが計算される。

【 0 0 2 6 】

【 発明が解決しようとする課題】上述した従来の加算回路は、縦続接続された2 段の4 入力NANDゲートを有する第1 型のCLAセルをさらに縦続接続して全体の桁上げを計算しているため、上記桁上げが伝播するクリティカルパスは少なくとも4 段の4 入力NANDゲートを含むため遅延時間が大きくなり、高速化の阻害要因となるという欠点があった。

【 0 0 2 7 】

【 課題を解決するための手段】本発明の加算回路は、それぞれm×n (m , n は整数) ビットの加数と被加数とから第1 階層の桁上げ発生関数および桁上げ伝播関数とを生成する第1 階層の桁上げ関数生成手段と、前記第1 階層の桁上げ発生関数と桁上げ伝播関数との供給を受けこれら第1 階層の桁上げ発生関数と桁上げ伝播関数とをmビット毎にブロック化した第2 階層のブロック桁上げ発生関数とブロック桁上げ伝播関数とを生成するn 個の第2 階層のブロック桁上げ関数生成手段と、前記第2 階層の桁上げ発生関数と桁上げ伝播関数との供給を受けこれら第2 階層の桁上げ発生関数と桁上げ伝播関数とをmビット毎にブロック化した第3 階層のブロック桁上げ発生関数とブロック桁上げ伝播関数とを生成するn / m 個の第3 階層のブロック桁上げ関数生成手段と、前記第3 階層のブロック桁上げ発生関数とブロック桁上げ伝播関数と最下位ビットに対する桁上げ入力である最下位桁上げ入力との供給を受けm-1 個の予め定めた第1 のビット番号の桁の桁上げである第1 の桁上げ出力を算出する第3 階層の第1 の桁上げ先見(CLA) セルと、前記第

7

1 の桁上げ出力と前記第2 階層の桁上げ発生関数と桁上げ伝播関数と前記最下位桁上げとの供給を受けそれぞれ $m-1$ 個の予め定めたビット 番号の桁の桁上げである第2 の桁上げ出力を算出する n/m 個の第2 階層の第2 のCLAセルと、前記第2 の桁上げ出力と前記第1 階層の桁上げ発生関数と桁上げ伝播関数と前記最下位桁上げ入力との供給を受け全ての桁の桁上げである第3 の桁上げ出力を算出する n 個の第1 階層の第3 のCLAセルと、前記第3 の桁上げ出力の供給を受け和を生成する和生成器とを備える加算回路において、前記第2、第3 のCLAセルが前記最下位桁上げ入力を伝播する最下位桁上げ伝播経路の遅延時間を前記第1 のCLAセルの前記遅延時間よりも小さくする構成の論理回路を備えて構成されている。

【 0 0 2 8 】

【 実施例 】 本発明の加算回路の一実施例を示す図1 を参照すると、この図に示す本実施例の加算回路は、従来例と同様の4 ビットのCLA加算器を用いたブロック構成の6 4 ビットの加算回路であり、従来と同様の第1 階層の桁上げ関数生成器1 と、第2 階層、第3 階層のそれぞれのブロック桁上げ関数生成器1 0 0 ~ 1 1 5、2 0 0 ~ 2 0 3 と、第3 階層のCLAセル3 0 0 と、和生成器2 とに加えて、第2 階層のCLAセル4 0 0 ~ 4 0 3、および第1 階層のCLAセル5 0 0 ~ 5 1 5 の代りに、それぞれ後述の第2 型のCLAセルであるCLAセル4 0 0 A ~ 4 0 3 A、および5 0 0 A ~ 5 1 5 A とを備える。

【 0 0 2 9 】 この発明を特徴づける上記第2 型のCLAセルの回路図を示す図2 を参照すると、この第2 型のCLAセルは、上述の第1 型のCLAセルの論理素子E 2、E 3、E 4、E 6、E 7、E 8、E 9、およびE 1 2に加えて、論理素子E 1、E 5、E 1 0、およびE 1 1 の代りに、関数P 0 ~ P 2 の否定論理積をとり論理値F 2 1 を出力する3 入力NANDゲートである論理素子E 2 1 と、関数P 1、P 2 の否定論理積をとり論理値F 2 2 を出力する2 入力NANDゲートである論理素子E 2 2 と、論理値F 2 1、F 2、F 3 の否定論理積をとり論理値F 2 3 を出力する3 入力NANDゲートである論理素子E 2 3 と、論理値F 2 2、F 6 の否定論理積をとり論理値F 2 4 を出力する2 入力NANDゲートである論理素子E 2 4 と、論理値F 2 3 と桁上げ入力C_i n との否定論理積をとり論理値F 2 5 を出力する2 入力NANDゲートである論理素子E 2 5 と、論理値F 2 4 と桁上げ入力C_i n との否定論理積をとり論理値F 2 6 を出力する2 入力NANDゲートである論理素子E 2 6 と、論理値F 2 5、F 4 の否定論理積をとり論理値C 0 を出力する2 入力NANDゲートである論理素子E 2 7 と、論理値F 2 6、F 7 の否定論理積をとり論理値C 1 を出力する2 入力NANDゲートである論理素子E 2 8 とを備える。

8

【 0 0 3 0 】 図2 において、関数P 0 が第3 階層のブロック桁上げ伝播関数PPP 1 6 に、P 1 がPPP 3 2 に、P 2 がPPP 4 8 に、関数G 0 が第3 階層のブロック桁上げ発生関数GGG 1 6 に、G 1 がGGG 3 2 に、G 2 がGGG 4 8 にそれぞれ対応している。

【 0 0 3 1 】 次に、本実施例の動作について説明する。

【 0 0 3 2 】 全体の加算回路の動作については、上述の従来例と同様であり、本発明に直接関連するもの以外は冗長としないよう説明を省略する。

【 0 0 3 3 】 まず、従来例で説明した式(1)、

(2)、(3)、および(4) でそれぞれ示される計算を実行し、第3 階層のブロック桁上げ発生／伝播関数GGG、PPP が算出され、これら関数GGG、PPP と桁上げC_i n の供給を受け、第3 階層のCLAセル3 0 0 が桁上げC 1 6、C 3 2、C 4 8 を計算するまでは、従来と同一である。

【 0 0 3 4 】 次に、CLAセル3 0 0 から供給される桁上げC 1 6 と関数GG 4、PP 4、GG 8、PP 8、GG 1 2、およびPP 1 2 とを用いて、第2 階層のCLAセル4 0 0 A が桁上げC 4、C 8、C 1 2 を計算する。同様に、CLAセル4 0 1 A ~ 4 0 3 A が、それぞれ桁上げC 1 6、C 2 0、C 2 4、C 3 6、C 4 0、C 4 4、C 5 2、C 5 6、C 6 0 を計算する。次に、第2 階層のCLAセル4 0 0 A ~ 4 0 3 A から供給される桁上げC 4、C 8、C 1 2、C 1 6、C 2 0、C 2 4、C 3 6、C 4 0、C 4 4、C 5 2、C 5 6、C 6 0 と関数G、P と桁上げ入力C_i n とを用いて、第1 階層のCLAセル5 0 0 A ~ 5 1 5 A が桁上げC 0、C 1、C 2、…を計算する。

【 0 0 3 5 】 これらCLAセル4 0 1 A ~ 4 0 3 A、およびCLAセルCLAセル5 0 0 A ~ 5 1 5 A は、上述のように、第2 型のCLAセルを用いる。この第2 型のCLAセルは、桁上げC 0 を算出する論理素子E 2 5、E 2 7 と、桁上げC 1 を算出する論理素子E 2 6、E 2 8 と、桁上げC 2 を算出する論理素子E 8、E 1 2 の各々が2 入力NANDゲートである。したがって、クリティカルパスとなる桁上げ入力C_i n の伝達パスに関しては、CLAセルの各々毎に2 入力NANDゲートの2 段構成となり、これが縦続接続されて全体の伝達パスは4 段の2 入力NANDゲートの縦続接続となる。

【 0 0 3 6 】 CLAセルを用いたこの種のブロック構成の加算回路は、下の階層すなわち第1 および第2 階層の上記CLAセルでは、桁上げ発生関数やブロック桁上げ発生関数、および桁上げ伝播関数やブロック桁上げ伝播関数が入力される時刻よりも、ブロック最下位への桁上げが入力される時刻の方が極めて遅くなる。一方、最上位階層すなわち第3 階層の上記CLAセルでは、桁上げ入力C_i n が入力される時刻よりも、ブロック桁上げ発生／伝播関数が入力される時刻の方が遅い。

【 0 0 3 7 】 一般に、CMOS 論理回路におけるNAN

50

Dゲートは入力数に対応する数のトランジスタの直列接続から構成されている。そのため、信号伝播時間について同一素子を用いた同一回路方式の2入力NANDゲートと4入力NANDゲートとを比較すると、前者の方が高速であり、したがって、下位からの桁上げ入力C_{in}が入力されてから上位への桁上げ出力が得られるまでの遅延時間は、4入力NANDゲートを用いた第1型のCLAセルよりも、2入力NANDゲートを用いた第2型のCLAセルの方が小さくより高速である。したがって、第1、第2階層において、第2型のCLAセルを用いることによりブロック最下位への桁上げ入力の遅延を低減でき高速化が図れる。一方、第3階層については、ブロック桁上げ発生／伝播関数に対してゲート段数が少ない第1型のCLAセルの方が高速化できる。(CMOS論理回路を構成する直列および並列トランジスタ接続における遅延時間については、例えば、ニール・HE・ウェステ他、CMOS VLSI設計の原理、丸善株式会社、昭和63年、第154～155頁(文献1)に記載されている。)例えば、0.8μmCMOSプロセスの場合、フアンアウト1のとき4入力NANDゲートおよび2入力NANDゲートの各々の遅延時間は0.35nS、0.25nSである。また、フアンアウト3のとき4入力NANDゲートおよび2入力NANDゲートの各々の遅延時間は0.50nS、0.35nSである。すなわち後者が前者に比し遅延時間が約30%小さいとされる。したがって、本実施例では、第2型のCLAセルを用いた2力NANDゲート4段の縦続接続により、

クリティカルパスである桁上げ入力C_{in}の遅延時間を約0.5nS低減、すなわち、約30%低減できる。

【0038】

【発明の効果】以上説明したように、本発明の加算回路は、第1、第2階層のCLAセルの最下位桁上げ伝播経路の遅延時間が第3階層のCLAセルの上記遅延時間よりも小さくする構成の論理回路を備えることにより、上記桁上げが伝播するクリティカルパスの遅延時間を低減し、演算の高速化が可能となるという効果がある。

10 【図面の簡単な説明】

【図1】本発明の加算回路の一実施例を示すブロック図である。

【図2】本実施例の加算回路における第2型のCLAセルの回路図である。

【図3】従来の加算回路の一例を示すブロック図である。

【図4】第1型のCLAセルの回路図である。

【符号の説明】

1 桁上げ関数生成器

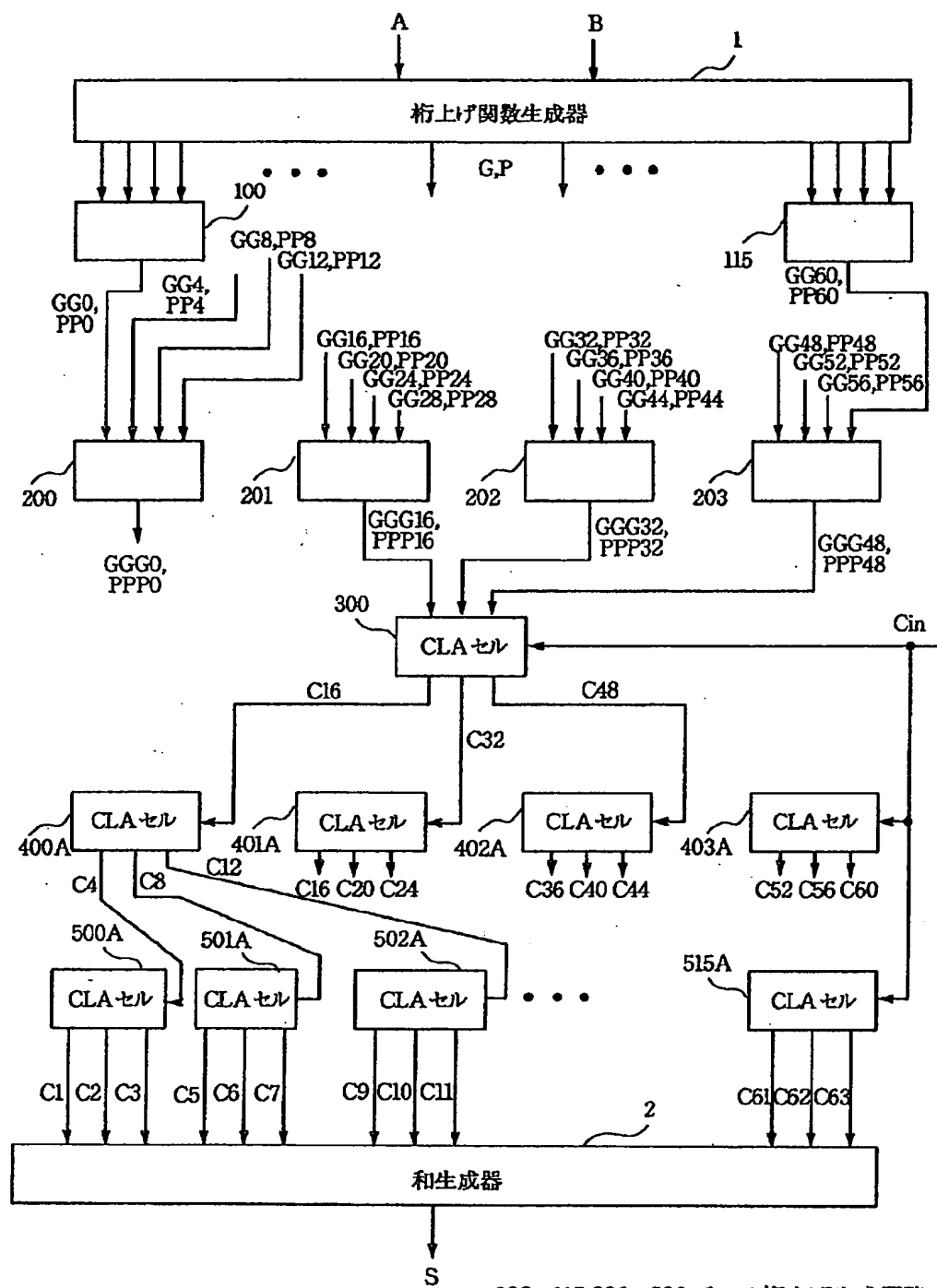
2 和生成器

100～115, 200～203 ブロック桁上げ関数生成器

300, 400～403, 500～515, 300A, 400A～403A, 500A～515A CLAセル

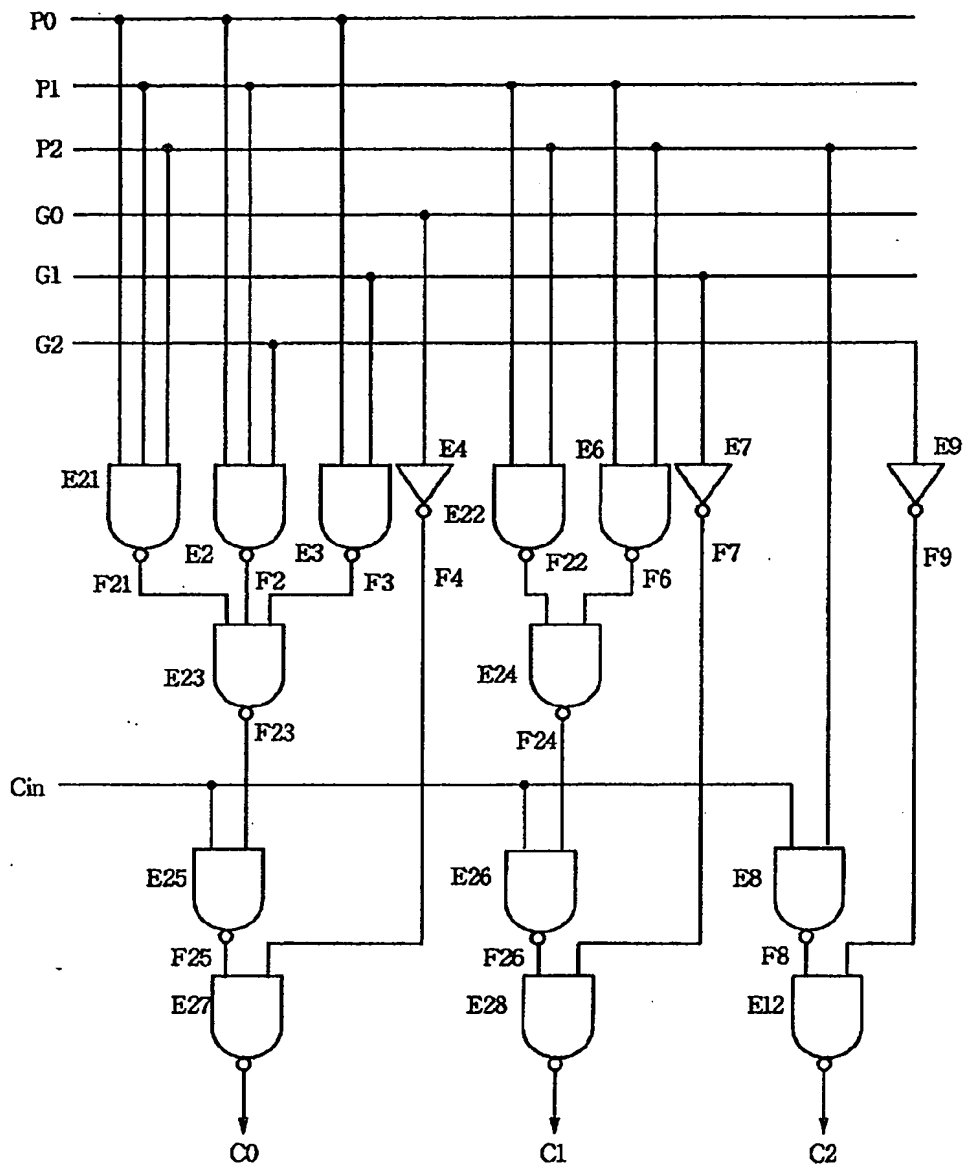
E1～E12, E21～E28 論理素子

【 図1 】

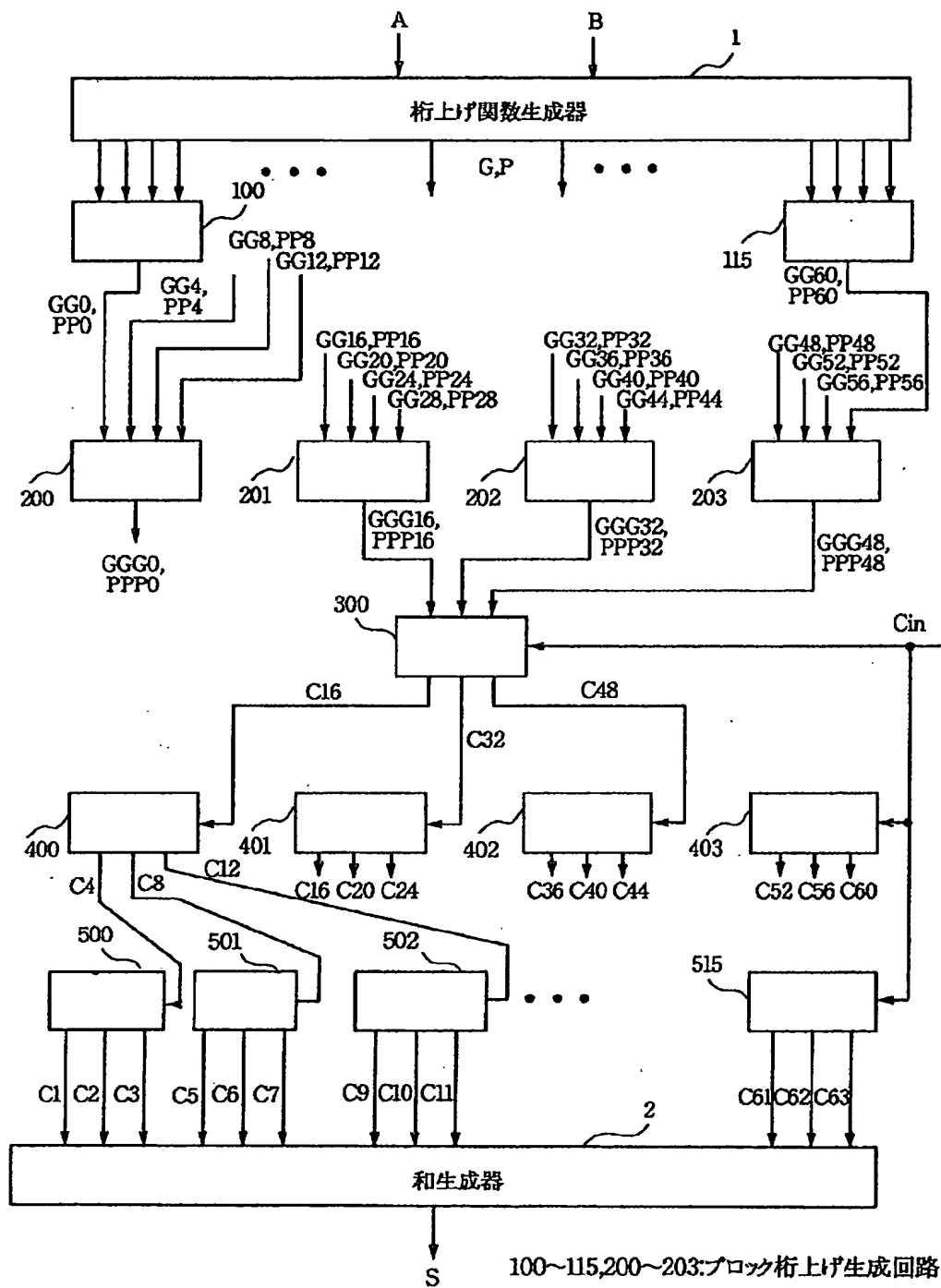


100~115, 200~203: ブロック桁上げ生成回路

【 図2 】



【 図3 】



【 図4 】

